The Path to Becoming a Software Architect

Image source: monument valley game

Have you ever wondered what career opportunities a developer has? What directions are open, beyond what horizons to grow. And most importantly, where are developers beyond the age of 45? Is there a developer among your friends who is over 45? I know several developers beyond this age, and many of them are hardcore programmers who even saw punch cards back in the day.

Translation: 你有没有想过开发人员有什么职业机会？什么方向是开放的，超越什么地平线才能成长。最重要的是，超过45岁的开发商在哪里？你的朋友中有没有超过45岁的开发人员？我认识几个超过这个年龄的开发人员，他们中的许多人都是铁杆程序员，他们甚至在过去看到过穿孔卡片。

There are several career paths a developer might take:

Translation: 开发人员可能会走以下几种职业道路：

● The first and obvious one is to grow in the area in which you are working. If you are a junior developer, then become to middle, then senior and lead roles.

● Transition to another technology stack. A significant number of developers moved into the mobile area when iOS and Android OS gained ground.

● Grow into a manager role. As a developer, the hugest staffing issue I saw was the shortage of competent managers. Smart managers are expensive, hence they are scarce. If the manager has a technical background, that will allow him to be on the same wavelength with the developers.

● Become a software architect. This direction will be considered in this series of articles.

● Get out of IT. Sometimes this happens. It is never too late to do what you like to do.

Translation: ● 离开IT。有时会发生这种情况。做你喜欢做的事永远不会太迟。

Article series

Translation: 文章系列

The Path to Becoming a Software Architect

Stakeholders in Software Architecture

Types of Software Architects

Quality attributes in Software Architecture

Documentation in Software Architecture

Translation: 软件体系结构中的文档

Certificates in Software Architecture

Translation: 软件体系结构证书

Books in Software Architecture

System Design Cheat Sheet

Yes, This Was My Path

In the past eight years, I have worked with Java EE, then moved to iOS, and became a team lead. I managed various developer teams, including Android, and Web stacks. Created the architecture of the network layer for several services developed by the company, with sockets and REST API. I became acquainted with the managerial role and the prospect of growing in this direction while in the position of team lead for over two years. In my next role, my goal is to grow as a software architect.

For most developers, the function of the architect on the project is often unclear, so in this series of articles, I will try to find the answers to these related questions. Who is an architect, what is the scope of responsibilities, and how to grow in this direction and outline an action plan for myself and beginners wanting to move along this path?

Translation: 对于大多数开发人员来说，架构师在项目中的作用往往是不明确的，因此在本系列文章中，我将尝试找到这些相关问题的答案。谁是建筑师，职责范围是什么，如何朝着这个方向发展，并为我自己和想要沿着这条道路前进的初学者制定行动计划?

Who Can Benefit from This?

Translation: 谁能从中受益?

This series of articles will help you if you belong to one of the following categories:

- IT developer or engineer. You are still growing as a developer, but you are looking ahead and planning your career. Even if the goals are initially vague, a person who consciously sets strategic goals will reach them much quicker than a person who does not plan where she is heading.

- Team leader, lead software engineer. You are at the highest stage of the software development discipline. To grow further, you have a choice to either learn one more stack of technologies, pursue a career outside software engineering, or to become a software architect.

- Software architect. You recently took this position, or have been working in this field for a long time. Perhaps one of the main qualities of such a specialist is the understanding that there are always areas that a person does not know and that the learning process is continuous.

- IT manager. Although you are a manager, you understand perfectly well that you should at least approximately understand what your subordinates or colleagues are doing. The acute problem of management is the technical incompetence of the manager in the field in which he or she is managing.

Translation: ● IT经理。尽管你是一名经理，但你非常清楚，你至少应该大致了解你的下属或同事在做什么。管理的尖锐问题是经理在他或她所管理的领域技术上的无能。

Who is an Architect?

Before moving on to more specific questions, it is necessary to define the software architect role and responsibilities.

A software architect is a software expert who makes high-level design choices and dictates technical standards, including software coding standards, tools, and platforms. The leading expert is referred to as the chief architect. (Wikipedia, The Free Encyclopedia, s.v. “Software architect”, https://en.wikipedia.org/wiki/Software_architect

Like most high-level positions, there are no clear criteria that define this role. However, it is possible to identify several responsibilities and qualities that contribute to the career of the architect.

First, let’s consider the characteristics of the architect:

● Communicability. Having talked with many software architects, I heard that it is one of the essential characteristics of this specialist. During the working day, they have to speak with customers in the language of business, managers of all levels, business analysts, and developers. If you have a natural charisma and you know how to convince people, then this will be a huge plus, as it is crucial to explain your actions correctly. Architects are laconic, eloquent, and competent speakers. The software architects with whom I spoke have highly developed skills in communication and persuasion. Another reason why this characteristic is most important is that the architect in this role participates in most discussion making processes, and often compromises must be reached that is acceptable and beneficial for all involved parties.

Translation: ● 传染性。在与许多软件架构师交谈后，我听说这是这位专家的基本特征之一。在工作日，他们必须用业务语言与客户、各级经理、业务分析师和开发人员交谈。如果你有天生的魅力，并且你知道如何说服别人，那么这将是一个巨大的优势，因为正确解释你的行为至关重要。建筑师简洁、能说会道、能言善辩。与我交谈过的软件架构师在沟通和说服方面有着高度发展的技能。这一特性最重要的另一个原因是，担任这一角色的架构师参与了大多数讨论过程，通常必须达成对所有相关方都可接受和有益的妥协。

● Broad and deep technical knowledge. It should be obvious since one cannot become a software architect with a medical background. Besides, the architect usually has expertise in several technological stacks at a decent level and should have a good understanding of a few other ones. The software architect should also be prepared to compose a large number of technical documentation, reports, and diagrams.

● Responsibility. You should understand that architect decisions are usually the most expensive. Therefore, a person in this position should take the most responsible approach to his work and the decisions made. If the developer’s error costs a couple of days of work of one person, then the architect’s mistake can cost person-years on complex projects!

● Stress resistance. You will have to make decisions because in this role, you will be asked to do so, and you will need a response. You will be working with different people from different areas, and you will have to deal with rapidly changing demands or even with changing business environments. Therefore, it is necessary to be ready for stress and to look for some ways to escape negative emotions. Work is always more pleasant when it brings pleasure. So if you choose this role only for the money, then think again.

● Management skills. This includes both organizational and leadership skills. The ability to lead a team, which may be distributed and composed of very different specialists, is essential.

● Analytic skills. Even if a specialist has a broad erudition in technology, he has tried many things on his own or participated in projects of various types. It does not guarantee that he can easily change the style of thinking to an architect. One of the most valuable tasks is the ability to represent an abstract problem in the form of some finite real object of the system, which developers are already evaluating, designing, and developing. Excellent communication skills are essential to represent the abstraction in the form of the final system to the members of the team and the customer. It will be necessary to communicate with both business and development, which is still to be done.

If we talk about the responsibilities of the architect, then here is the perfect example from the 19th century about bridge construction. At that time, the tests of the newly constructed bridge were the following: the key group of engineers, architects and workers stood under the bridge while the first vehicles were on it. Thus, they staked their lives upon the construction and the strength of the structure. So if there is a question — what is the responsibility of the software architect on the project? The answer is, he is responsible for everything.

Translation: 如果我们谈论建筑师的责任，那么这里是19世纪关于桥梁建设的完美例子。当时，新建桥梁的测试如下：当第一批车辆在桥上时，关键的工程师、建筑师和工人站在桥下。因此，他们将生命押在了结构的施工和强度上。所以，如果有一个问题——软件架构师对项目的责任是什么？答案是，他对一切负责。

If you give up loud and beautiful phrases, then the architect’s work includes:

● Identifying the stakeholders on the project.

● Identifying business requirements and requirements of the stakeholders on the project.

● Designing the entire system based on the received requirements.

● Choosing the system architecture and each component of this system at a high level.

● Choosing the technologies for the implementation of each component and connections between the parts.

Translation: ● 选择用于实现每个组件的技术以及部件之间的连接。

● Architectural review. Yes, yes, it exists.

Translation: ● 建筑审查。是的，是的，它是存在的。

● Code-review.

● Writing project documentation and support it.

● Creating unified development standards in the company.

● Controlling the architecture during the next iteration of the system release.

Translation: ● 在系统发布的下一次迭代中控制体系结构。

It is only a subset of the software architect's responsibilities. The most important responsibility is full technical support of the project from the moment of inception, through product release, to the development of enhancements. And supporting the next releases. It will be necessary to switch a lot between different tasks during the working day.

Translation: 这只是软件架构师职责的一个子集。最重要的责任是从项目开始的那一刻起，通过产品发布，到增强功能的开发，为项目提供全面的技术支持。并支持下一个版本。在工作日中，有必要在不同的任务之间进行大量切换。

How to Become a Software Architect?

To begin with, it is requisite to define milestone goals that lead to achieving your strategic goal of becoming a software architect. For me, such goals for the next six months are:

● Understand and try several technological stacks. My current knowledge is concentrated in the field of iOS. It is necessary to try Android, several server languages, to start python, and refresh Java EE skills. The architect is a full-stack developer, so it is essential to have a broad technical knowledge.

● Reading literature. It is required to determine the most valuable books and articles that will help to grow in this direction. Usually, the most effective way to find such literature is to ask other professionals in this field for their recommendation. In one of the future articles, I plan to give you such a list of literature.

● Find a mentor. It is desirable to find a software architect at your current place of employment. It is always easier to get experience from a trained specialist than to start considering a particular area from scratch. It is requisite to be prepared to ask the right questions from your mentor.

Translation: ● 找一个导师。最好在你目前的工作地点找到一位软件架构师。从受过训练的专家那里获得经验总是比从头开始考虑某个特定领域更容易。有必要准备好向你的导师提出正确的问题。

● Study courses/obtain certificates. There are many courses and certifications available, but only a few are worth their money, and the higher level courses cost a lot of money. I have attended the architectural courses of Luxoft (http://www.luxoft-training.com/it-course/ARC-001/), which have proven to be a worthy investment. It is crucial that the lecturer of the course be a professional in this field and be able to answer questions. As for certificates, before starting, it is best to understand whether there are authoritative certification systems for architects and whether it is worthwhile obtaining the certification. This point I will discuss in a future article of this series.

One of the essential parts is a clear and stable plan review. What has been done, what should be reviewed, and where to accelerate or which goal to remove as useless.

Check Your Readiness Level

If you are interested in this introductory article from the series on how to become a software architect, or if you suddenly have thoughts to try this path, it is worth making sure that you want it.

Firstly, people are afraid of everything new. A new position, new kind of stress, as opposed to the comfortable status quo. Of course, the choice is not always unambiguous and depends on how much you are willing to change something in your life. At the same time, it can depend not only on you but also on the family, your financial commitments, parents, and other factors.

Translation: 首先，人们害怕一切新事物。一个新的位置，一种新的压力，与舒适的现状相反。当然，选择并不总是明确的，取决于你愿意在多大程度上改变你的生活。同时，这不仅取决于你，还取决于家庭、你的经济承诺、父母和其他因素。

Secondly, this path takes several years. The process of becoming a software architect does not happen overnight. As a team lead, I realized what to do and how to deal with stress only a year after I was appointed to an official position. At the same time, six months before that, I performed it unofficially. One software architect I know said that he understood what his responsibilities are 18 months after he was promoted to this role. Such intervals of time are reasonable, and you need to know whether you are ready to move in this direction. Even if you do not have a stable plan available, it is better to start taking small steps that move you ahead, rather than remaining in the same place.

Standing in the same place in IT is a synonym for stagnation and personal fetters in life.

Recommended Books

Software Architecture in Practice (3rd Edition) (SEI Series in Software Engineering)

Essential Software Architecture